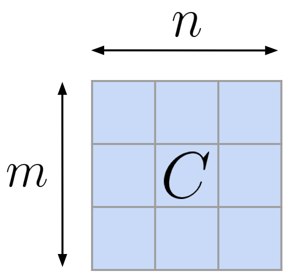
Applying Optimal Transport to recommendation

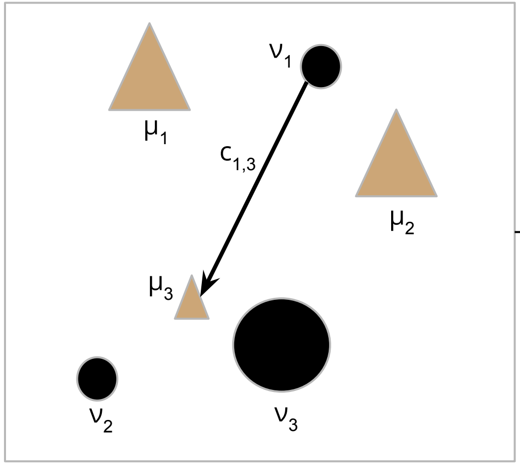
Théophane Vallaeys
Lucas Marandat
Guillaume Kunsch

(Group MatrixT)

Optimal Transport (OT): a simple approach

Knowing the cost matrix:
(cost of moving from μ_i to ν_j)





C

Goal

$$\min_{\pi \in \Pi(\mu,\nu)} \langle C, \pi \rangle$$

Using the SK algorithm

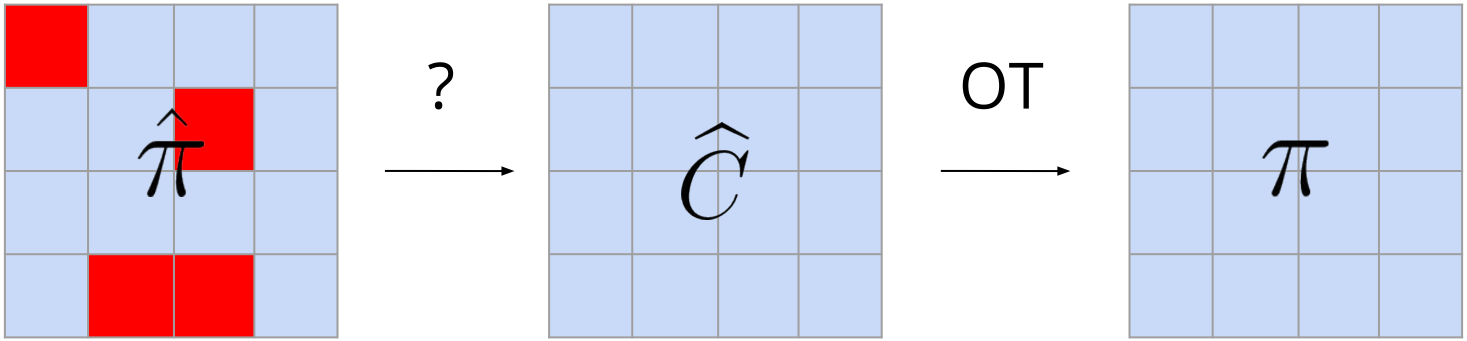
π

	ν_1	ν_2	ν_3
μ_1	0.2	0.2	0
μ_2	0	0	0.4
μ_3	0	0	0.2

π

From OT to Inverse OT

Settings of OT don't match what we have in practice: **C is unknown**

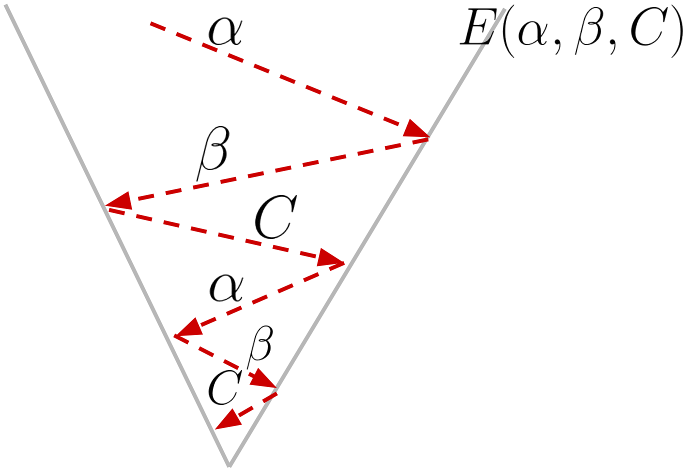


Completing $\hat{\pi}$: $$\min_{C} KL(\hat{\pi}||\pi^{C})$$ → $$\min_{\alpha,\beta,C} E(\alpha, \beta, C)$$

Learning Cost Function for Optimal Transport, Ma et al., 2021

<u>Algorithm: repeat until convergence</u>

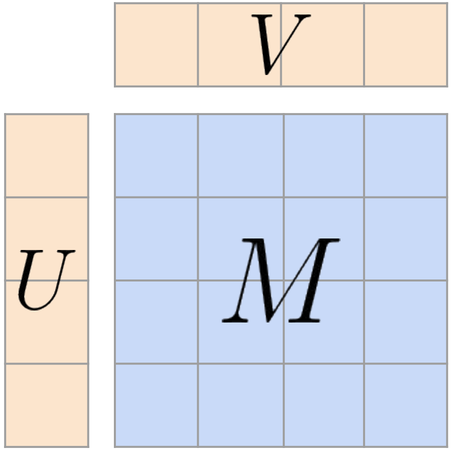
$$\alpha \leftarrow \min_{\alpha' \in \mathbb{R}^n} E(\alpha', \beta, C)$$
$$\beta \leftarrow \min_{\beta' \in \mathbb{R}^m} E(\alpha, \beta', C)$$
$$C \leftarrow \min_{C' \in \mathbb{R}^{n \times m}} E(\alpha, \beta, C')$$



Cost parametrization:

$$C_A = U^\top A V$$
$$A \in \mathbb{R}^{k \times k}, k < n, m$$

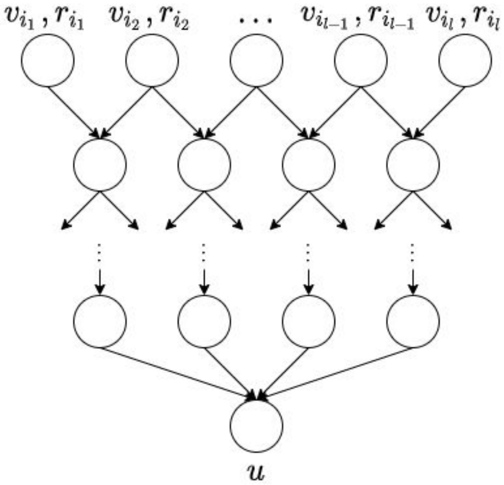
$U \in \mathbb{R}^{n \times k}, V \in \mathbb{R}^{m \times k}$ users/movies features



Multiple ideas to extract features

$$U, V$$

Movie tags



Learning:

Recommendation Matrix

$$M \in \mathbb{R}^{m \times n}$$

Extract features

$$\min_{U,V} \|UV^T - M\|$$

Normalization

$$\hat{\pi} = \frac{M}{\sum_{i,j} M_{ij}}$$

U, V

$\hat{\pi}$

IOT Algorithm

$$\min_{C_A} KL(\hat{\pi} \| \pi^{C_A}) \longrightarrow C_A$$

where π^{C_A} is a solution of the OT with cost C_A

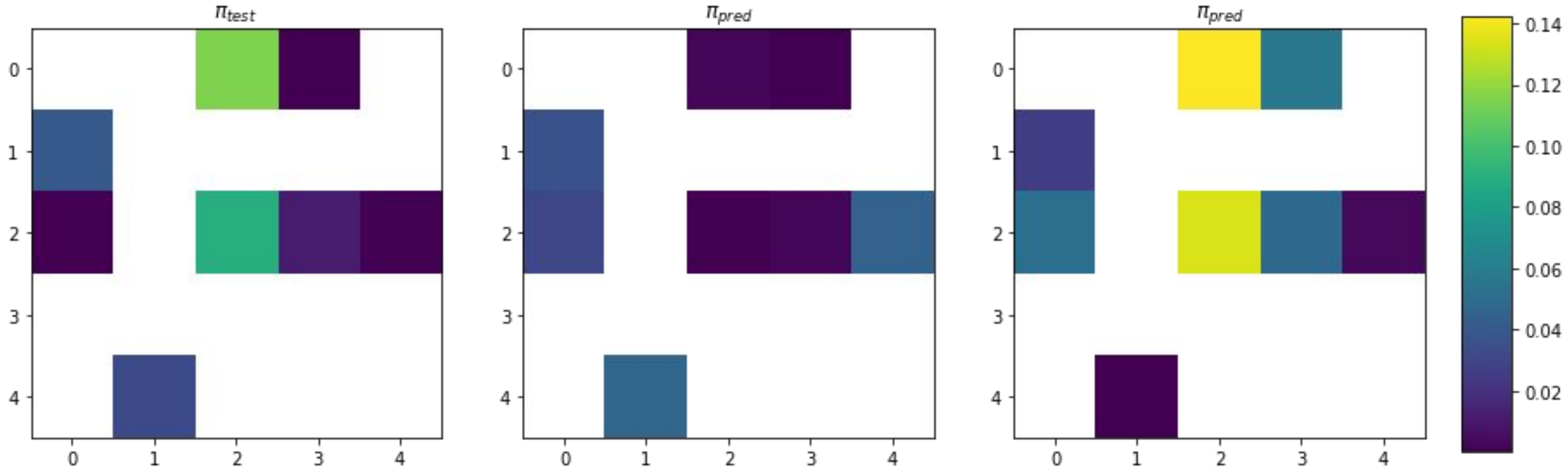
Predict: for user i and movie j

OT Algorithm

$$\xrightarrow{C_A} \min_{\pi} \langle C_A, \pi \rangle - \epsilon H(\pi) \xrightarrow{\pi} F(\pi_{ij})$$

with F a rescaling function

Preliminaries results:



Test Data

Matrix Factorization
(RMSE=0.056)

Inverse Optimal Transport
(RMSE=0.037)